Claims.

1. A method for downloading data, comprising the steps of:

establishing a connection over a communication network between a remote client and a server, said connection comprising a socket;

receiving a download request from said client via said connection for download of information from said server;

transferring said socket to a download manager process executing on said server; and

transmitting said information to said client from said download manager process using said socket.

2. The method according to claim 1, further comprising the step of converting said socket to a non-blocking socket.

3. The method according to claim 1, wherein said step of transferring said socket comprises the steps of:

constructing a copy of said socket; and

thereafter closing said socket; and

transferring said copy to said download manager process to define a second connection between said download manager process and said client using said copy.

4. The method according to claim 1, wherein said client comprises a plurality of clients, and said steps of establishing a connection, and transferring said socket, and are performed substantially concurrently with respect to each of said plurality of clients.

5. The method according to claim 1, wherein said download request further comprises an indication of a file on said server, further comprising the steps of:

enqueueing said download request in said download manager process with other download requests concurrently being managed therein.

6. The method according to claim 1, wherein said connection comprises a TCP session.

7. The method according to claim 1, wherein said socket of said connection is a blocking socket.

8. The method according to claim 1, wherein said server allows a maximum number of open file descriptors, further comprising the steps of:

spawning a duplicate download manager process when said maximum number of open file descriptors is exceeded;

performing said step of receiving a download request in one of said download manager process and said duplicate download manager process; and

servicing previously pending requests in another of said download manager process and said duplicate download manager process.

9. The method according to claim 1, further comprising the steps of:

accepting new connections for additional clients over said communication network;

establishing respective first threads in said server to operate said connection and said new connections;

establishing a second thread to operate said download manager process; and

communicating between said download manager process and said client and said additional clients via said connection and said new connections respectively using non-blocking I/O.

10. A computer software product, comprising a computer-readable medium in which computer program instructions are stored, which instructions, when read by a computer, cause the computer to perform a method for downloading files from said computer over a data network comprising the steps of:

intercepting a download request for information that is received via a first connection from a remote client, said first connection comprising a socket;

installing a download manager in said computer;

transmitting a set of data comprising said download request and a descriptor of said socket to said download manager to define a second connection between said download manager and said client using said descriptor; and

downloading said information from said computer to said client via said second connection.

11. The computer software product according to claim 10, wherein said client comprises a plurality of clients, and said steps of intercepting a download request, and transmitting a copy of said socket, and downloading said information are performed substantially concurrently with respect to each of said plurality of clients.

12. The computer software product according to claim 10, wherein said computer is further instructed to perform the steps of

after performing the step of transmitting a set of data closing said first connection; and

converting said socket to a non-blocking socket.

13. The computer software product according to claim 10, further comprising the steps of:

enqueueing said download request in said download manager with other download requests concurrently being managed therein.

14. The computer software product according to claim 10, wherein said first connection comprises a TCP session.

15. The computer software product according to claim 10, wherein said socket of said first connection is a blocking socket.

16. The computer software product according to claim 10, wherein said computer allows a maximum number of open file descriptors, further comprising the steps of:

spawning a duplicate download manager when said maximum number of open file descriptors is exceeded;

receiving said set of data in one of said download manager and said duplicate download manager; and performing said step of download information for previously pending requests in another of said download manager and said duplicate download manager.

17. The computer software product according to claim 10, wherein said computer is further instructed to perform the steps of:

accepting new connections for additional clients ;

establishing respective first threads in said server to operate said connection and said new connections;

establishing a second thread to operate said download manager process; and

communicating between said download manager process and said client and said additional clients via said connection and said new connections respectively using non-blocking I/O.

18. A system for downloading information over a data network, comprising:

a server connectable to a plurality of clients across said data network via blocking sockets, said server being adapted to intercept download requests from said clients, and to associate each of said download requests with respective copies of said blocking sockets; and

a download manager executing in said server that receives said download requests and said copies from said server, said download manager being adapted to convert said copies to non-blocking sockets, said server thereupon closing said blocking sockets, wherein said download manager causes said download requests to be serviced from said server across said data network via respective ones of said non-blocking sockets.

19. The system according to claim 18, wherein said download manager is a subassembly of said server.

20. The system according to claim 18, wherein said download manager comprises a queue for holding said download requests, wherein said download requests are serviced in turn from said queue.

21. The system according to claim 18, wherein said server establishes respective first separate threads to operate sockets connecting said server with said clients and establishes a second thread to operate said download manager, said download manager converting said blocking sockets to non-blocking sockets and communicating with said clients via said non-blocking sockets using non-blocking I/O.